REMARKS

The final Office Action of January 4, 2006, considered and rejected claims 1-24. Claims 1, 19, 20 and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bauermeister et al (U.S. Patent No. 5,586,241). Claims 2-17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bauermeister et al (U.S. Patent No. 5,586,241) in view of Rappoport (WO 98/36630). Claim 18 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bauermeister et al (U.S. Patent No. 5,586,241) in view of Betrisey et al (U.S. Patent Publ. No. 2001/00448764). Claims 21 and 22 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bauermeister et al (U.S. Patent No. 5,586,241) in view of Brassell et al (U.S. Patent No. 5,684,510). Claim 24 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bauermeister et al (U.S. Patent No. 5,586,241) in view of Qureshi et al (U.S. Patent No. 6,456,305).[1]

By this paper, claims 1, 19 and 20 have also been amended, and no claims have been added or cancelled.[2] Accordingly, following this paper, claims 1-24 remain pending. Of these claims, claims 1, 19, and 20 are the only independent claims at issue.

As reflected in the claims listing above, claim 1 is generally directed to a method for accessing a scaled font and synthesizing a font variant. As recited, this method includes accessing a font file having a plurality of glyphs and standard instructions applicable for constraining each of the plurality of glyphs, and wherein each of the plurality of glyphs stores glyph features. Further, as recited in claim 1, the method includes utilizing the font file in accessing a scaled font that has been scaled for rendering at a target size and target resolution, wherein the font file includes enough hints to synthesize the scaled font. Claim 1 also recites accessing one or more external font parameters that alter how glyphs of the scaled font should be

[1] Although the prior art status and some of the assertions made with regard to the cited art is not being challenged at this time, Applicants reserve the right to challenge the prior art status and assertions made with regard to the cited art, as well as any official notice, which was taken in the Office Action, at any appropriate time in the future, should the need arise, such as, for example in a subsequent amendment or during prosecution of a related application. Accordingly, Applicants' decision not to respond to any particular assertions or rejections in this paper should not be construed as Applicants acquiescing to said assertions or rejections.

[2] Support for the claim amendments are clearly supported by paragraphs [0030], [0035], [0036], [0049] and [0054], among other passages throughout the specification. Accordingly, it is respectfully submitted that the amendments to the claims do not add new matter, and entry thereof is respectfully requested.

rendered and applying the one or more external font parameters to the scaled font to synthesize a font variant such that hints from the scaled font are preserved in the font variant.

Claims 19 is directed to a similar method being performed by a computer program product, while claim 20 is directed to a computing system configured to perform a similar method.

While Bauermeister is generally directed to methods and systems for parametrically generating characters and fonts, Applicants respectfully submit that the cited art fails to make obvious the claimed invention. For example, the cited art fails to disclose or suggest, a font file is accessed which includes a plurality of glyphs and standard instructions for constraining each of the plurality of glyphs, and in which the font file includes hints sufficient to render the scaled font. (claims 1, 19 and 20). In addition, Bauermeister fails to disclose or suggest that the one or more external font parameters include font-hinting instructions in the same language as the hints of the scaled font language (claim 25).

Specifically, Bauermeister discloses a Terafont font generation technique that is run on a computer and allows a user to create, specify, replicate, generate and/or supply scalable fonts. (Abstract; Col. 4, ll. 19-22). The Terafont system allows the character outline of one or more characters within a font to be built by a font engine that executes each of two files which contain Terafonts and parameters, respectively. (Col. 10, ll. 27-30). In particular, a first, universal font generation file includes a set of universal font generation rules which provides consistent rules usable with any font within a single alphabet (the Terafont source file). (Abstract; Col. 9, ll. 25-29, 39-42). The Terafont file includes (i) instructions for computing global variables; (ii) instructions to build glyphs; (iii) a character map identifying characters within the fonts; and (iv) global constants used to specify the fonts. (Col. 4, ll. 31-40). Before the file is used, the Terafont source file is compiled into binary form to create a Terafont binary file. (Col. 10, ll. 58-62).

Bauermeister also teaches that the Terafont system uses a second file which includes measurement data and various other parameters to visually characterize a font (the parameter data source file). (Col. 4, ll. 41-44; Col. 10, ll. 48-57). The parameter data source file includes a Name section which specifies the font name and family, a PANOSE classification number, and another section used to override default variables for each glyph within a font. (Col. 20, ll. 26-

34). Similar to the Terafont source file, the parameter data source file is compiled into binary form, and a parametric data file is created. (Col. 10, ll. 62-64).

The method of generating a Terafont glyph or set of glyphs for display on an output device is described in Bauermeister with reference to Figure 6. In the described method, Bauermeister teaches an initial step of requesting both the Terafont binary data file (i.e., global data) and parametric data file (i.e., font specific data). (Col. 23, ll. 60-62). Thereafter, constants within the parametric data are stored in memory and global values are computed as defined by the universal font generation rules. (Col. 23, ln. 64 to Col. 24, ln. 2). A font engine subsequently determines if a parametric data value is to override a default value and, if so, such an override occurs. (Col. 24, ll. 3-7). A desired character or series of characters is then located in the Terafont binary data file's character map and the functions to produce the outline for the glyph are executed. (Col. 24, ll. 10-19).

Accordingly, Bauermeister appears to teach a font system in which two files are required to generate a font, and fails to teach or suggest that a single font file that includes both a plurality of glyphs and standard instructions for constraining each of the plurality of glyphs and also includes hints sufficient to render a scaled font, as claimed in combination with the other recited elements (claims 1, 19 and 20).

In fact, Bauermeister expressly teaches that no single file contains sufficient information to generate any glyph. For example, as described above, the Terafont binary file provides global variables and a character map, while the parametric data file provides font specific measurements used to generate the font. Stated another way, Bauermeister appears to teach that only the Terafont binary file includes a plurality of glyphs (i.e., the character map); however, this file is only alphabet-specific and does not generate any particular font. Thus, measurement and other data from the parametric data file is required for font generation. In fact, Bauermeister expressly teaches the opposite as "characteristics of individual characters comprising a specific font are defined to fully characterize the outlines of the characters in the font" when the font parameters are "combined with the universal font generation rules." (Col. 4, ll. 41-44). Stated another way, both the Terafont binary data and the parametric binary data, from their respective files, "are required for generating an output font." (Col. 23, ll. 60-62).

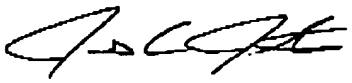
Moreover, Bauermeister expressly teaches away from the use of a single file which includes both standard instructions and a plurality of glyphs and which is sufficient to synthesize a scaled font. For example, Bauermeister notes that conts which are represented by scalable outlines include "all points, hint programs, and metric data are stored in the data file." (Col. 2, ll. 21-40). Bauermeister goes on to criticize such a data file because the file size requires 35-70 kBytes of data. (Col. 2, ll. 21-40). In fact, the size of such files appears to be a primary motivation for creating Bauermeister using separate files. In particular, the Terafont file stores universal rules such that it can be used with multiple fonts of the same alphabet. This enables "only a few input parameter values...to be stored in the binary parameter files defining a font." (Col. 3, ln. 58 to Col. 4, ln. 4). Consequently, the binary parameter files would likely be small in size, e.g., requiring less than two kBytes of hard disk storage space for each font." (Col. 4, ll. 1-4).

Accordingly, for at least these reasons, Applicants respectfully submit that the independent claims are allowable over the cited references. Moreover, inasmuch as each independent claim is allowable over the cited references, it will also be appreciated that all other rejections of record with respect to the dependent claims are now moot and therefore need not be addressed individually.

Thus, Applicants respectfully submit that the pending claims (1-25) are in condition for immediate allowance over the art of record. In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 3 day of March, 2006.

Respectfully submitted,



RICK D. NYDEGGER
Registration No. 28,651
JENS C. JENKINS
Registration No. 44,803
Attorneys for Applicant
Customer No. 047973

RDN:JCJ:ccn
PPA0000002645V001